

Smart Wire User Guide



Version 4.2.2 Build 8

July 27, 2007

Smart Wire is owned by Varsity Systems, LLC and is distributed under special agreement with WoodWing Software, The Netherlands and its distributors worldwide.

Software License Agreement

This is a legal agreement between you and Varsity Systems, LLC covering your use of Smart Wire (the “Software”). Be sure to read the following agreement before using the Software. BY USING THE SOFTWARE (REGARDLESS IF YOU HAVE REGISTERED THE SOFTWARE OR NOT), YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE AND DESTROY ALL COPIES IN YOUR POSSESSION.

The Software is owned by Varsity Systems, LLC and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the Software like any other copyrighted material (e.g., a book or musical recording). Paying the license fee allows you the right to use one copy of the Software on a single computer for one wire service. You may not use it on more than one computer at the same time. You may not rent or lease the Software, nor may you modify, adapt, translate, reverse engineer, decompile, or disassemble the Software. If you violate any part of this agreement, your right to use this Software terminates automatically and you must then destroy all copies of the Software in your possession.

The Software and its related documentation are provided “AS IS” and without warranty of any kind and Varsity Systems, LLC expressly disclaims all other warranties, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Under no circumstances shall Varsity Systems, LLC be liable for any incidental, special, or consequential damages that result from the use or inability to use the Software or related documentation, even if Varsity Systems, LLC has been advised of the possibility of such damages. In no event shall Varsity Systems, LLC liability exceed the license fee paid, if any. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitations may not apply to you.

This Agreement shall be governed by the laws of the State of Michigan, USA. If for any reason a court of competent jurisdiction finds any provision of the Agreement, or portion thereof, to be unenforceable, that provision of the Agreement shall be enforced to the maximum extent permissible so as to effect the intent of the parties, and the remainder of this Agreement shall continue in full force and effect.

Introduction

Smart Wire is a program designed to capture text data from a wire service modem such as those supplied by the Associated Press or similar services. This data is parsed and separated into files according to the wire service's conventions and saved into a WoodWing Smart Connection Enterprise database or to a series of folders on a file server as "stories". The "stories" are saved as InCopy documents.

Setting up this program will take some knowledge of the wire service conventions, the Smart Connection Enterprise system, and an applicable work flow concept.

Installation

System and Hardware Requirements

Smart Wire is available in Macintosh (OSX) or Windows (98 or newer) versions. In either case RAM and hard disk system requirements are minimal,

Wire service modems have serial RS-232 ports, normally with 25 pin D-style connectors. If the computer you wish to use does not have a serial port (most Macs do not) then you can purchase a USB to serial converter. We recommend one of the models from Keyspan (www.keyspan.com). You will find the devices under Serial Port Products and they come in 1, 2, and 4 port models. All should work fine with Smart Wire.

Software Installation

Simply copy the Smart Wire application file along with your required Script file (described later) into an appropriate folder on your machine..

After you have run Smart Wire for the first time you will see a preference file in the same folder as the application called ***Smart WirePrefs***. Do not delete or move this file or you will need to set up all the preferences again.

One copy of Smart Wire will capture data from one serial port i.e. one wire service. Multiple wire services can be captured on the same computer using multiple copies of Smart Wire. To accomplish this each copy of Smart Wire should reside in a separate folder and will require its own serial port.

Authorization

When you purchase Smart Wire, you will be provided with an authorization code. The first time you start Smart Wire a dialog box will appear asking you for this code. Type it in exactly as you receive it and Smart Wire will then be authorized on the computer you are using.

If you ever need to move Smart Wire to another machine you should first UnAuthorize it using the button on the About Smart Wire window. This will free your serial number to be installed elsewhere.

Configuration

Concept

A typical hardware configuration for wire capture is fairly simple and has several components:

- A database server running the Enterprise server software or a file server for a folder based system such as Smart Connection Pro or Smart Connection Light.
- A computer running Smart Wire (or several instances of Smart Wire) connected to the wire service modem via a serial cable and possibly a Keyspan device.
- All computers tied together with a local area network.

The system, or software, configuration is a bit more complicated, but basically involves tying the wire service's story information into the workflow you have set up for Enterprise or a folder based system. This will allow the user to easily find the type of stories they are looking for. There is much more detail on this subject in the following sections.

Computer Setup

A connection to the file server will need to be established if files are to be saved this way. The Enterprise connection will be established in the Enterprise Setup section below.

The computer should be set up so that Smart Wire starts up after a re-boot in case of a power outage although it makes good sense to use an uninterruptable power supply for this machine as well as for the servers. (See section on Preferences.)

Modem Connection

As mentioned previously, you will need to connect the wire service modem to a serial port on the computer. Appropriate cables are normally available from most computer stores, mail order houses or may have been provided by your wire service. You may need to include a null modem in that connection. If your computer does not have a serial port then refer to System and Hardware Requirements found on the previous page.

The Anatomy of a Wire Story

All wire services start their stories with a header section. This header contains information about story.

The most important item is the "category" of the story which is assigned by the service. Each wire service has their own method of categorizing, but there are some standards. Your wire service company should be able to provide a complete list. Usually these categories are specified by one letter, a-z, and make some kind of sense such as "s" for sports, "f" for financial, etc. Your job is to connect these category codes with your workflow within either an Enterprise database or a folder system. The specifics for each follow.

Another key element of a wire story is the slug line which is really the title of the story. This gives a few key words that should give you enough information to determine what the story is about.

Other elements of importance are:

- Priority. One character denoting the importance of the story.
- Length. The wire services usually give this in number of words. Smart Wire will also store length in total characters.
- Time and Date. Smart Wire will stamp each story with this as it comes in.

Enterprise Configuration

The key to using wire service information is to be able to find what you want from the many stories sent each day. Primarily the category is used for a gross division of stories into areas of interest to different sections of a publication such as Sports or Financial.

Wire service work flows can be done many ways, but we recommend that all wire stories have read only privileges so that they are not inadvertently changed. When a story is found that a user wishes to use, it is a simple process to copy the story into the actual publication and issue for editing leaving the original intact.

We suggest that you set up your Enterprise system with a separate Publication specifically for wire stories perhaps called “Wire”.

You will then want to create Sections in Enterprise corresponding to the categories in the wire service. As you will see in the Preferences section, you can direct many wire service categories into one Enterprise section. So setting up Sections will take some thought and knowledge of how your wire service categorizes and how you want your workflow to function.

Next you will need to create States for the Publication. Our recommendation is the wire publication be read-only so only one state needs to be created. It could be called “Original” or something similar.

The Authorizations would be set up as we suggest to allow editorial users read only privileges for this Publication.

In the Enterprise system the next level under Publication is Issue. Normally new issues are created dynamically for each publication date. In this case, however, we suggest that you create a permanent Issue for each separate wire service assuming that you have more than one. Since an issue is required in the Enterprise system, you will still need to create one even if you only have one wire service.

Note that you could have a separate Publication for each wire service, but the only reason for doing this would be if the categorization is radically different for the each wire services but this is uncommon

Smart Connection Enterprise

Publication: WIRE

[Edit](#)

Issues

7	AFP
8	Associated Press
9	Notimex

[New](#)

Sections

12	General
13	State
14	Sports
15	Financial

[New](#)

States

24	Article	Original
--------------------	---------	----------

[New](#)

Authorizations

11	admin	VRWDCK
12	copy	VR
13	page	VR

[New](#)

An example of a simple Enterprise configuration for multiple wire services

Folder Configuration

Again, the key to using wire service information is to be able to find what you want from the many stories sent each day. Category is typically used for a gross division of stories into areas of interest and different sections of a publication such as Sports and Financial.

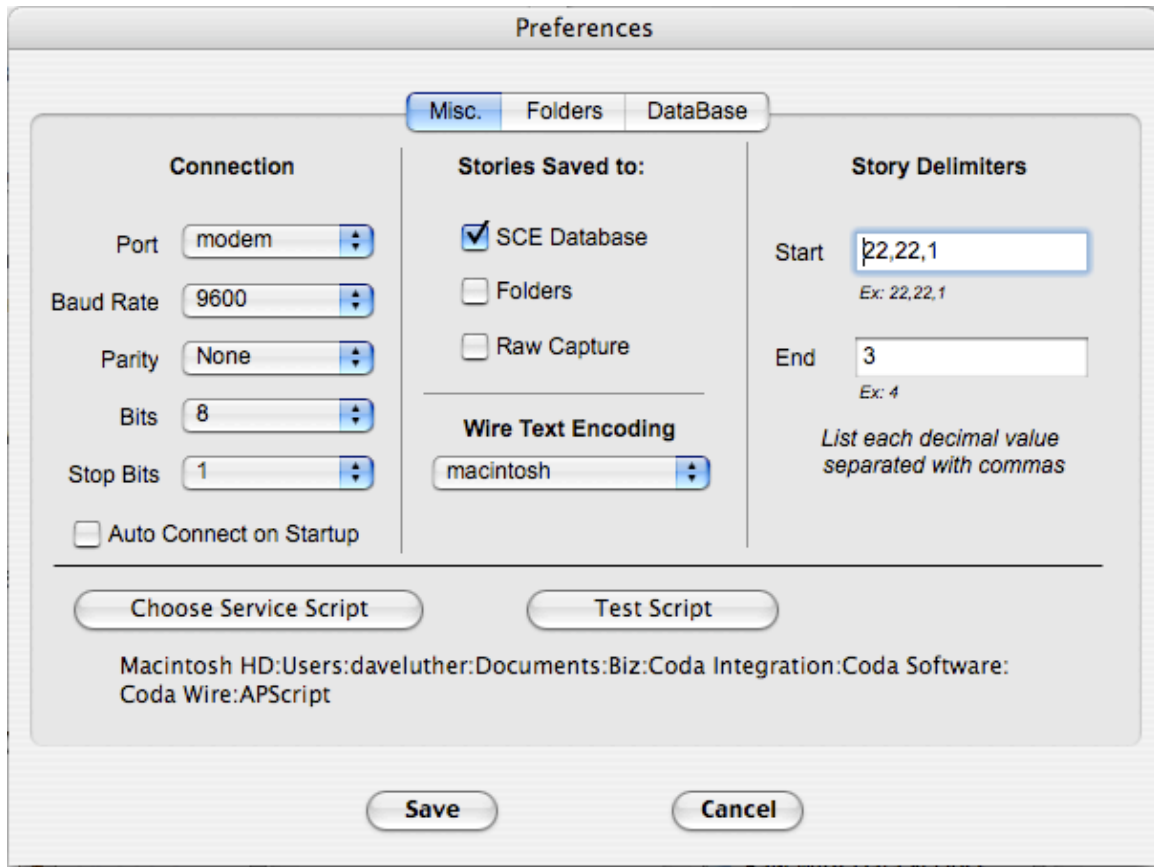
In a folder system you will create a folder on the designated file server, probably called “Wire” or named for the wire service.

Within this folder you will need to create sub folders for each category of interest. As you will see below, you can direct many wire service categories into one sub folder. So setting up these folders and the preferences will take some thought and knowledge of how your wire service categorizes and how you want your workflow to function.

Wire service work flows can be done many ways, but we recommend that all wire stories have read only privileges so that stories are not inadvertently written over. When a story is found that a user wishes to use it is a simple process to copy the story into the actual publication and issue for editing leaving the original intact.

Preferences

Preferences will be located under the menu that normally contains this item based on the operating system you are using. When you invoke that menu you will see the following:



This is the first of three preference screens, Miscellaneous (Misc.). The others are brought up by clicking on their title tabs: Folders, Database.

At any point you can click on the Save button to save all of the changes you have made and dismiss the Preference dialog. You can also click on the Cancel button which will dismiss the dialog with no changes made to the preferences.

You cannot connect to the wire service and thus receive stories while the Preferences window is open.

Misc.

As the title suggests this is where many different things are set. Each distinct group of settings has a heading and is separated by space or lines or both.

Connection

This is where the serial port is configured.

The Port popup will contain a list of all the serial ports available on your machine. Set this popup to the port connected to the wire service modem.

The remaining pop-ups, Baud Rate, Parity, Bits, Stop Bits, should be set according to the wire service specifications for their modem. Commonly Baud Rate is 9600, Parity is None, Bits are 8, Stop Bits are 1.

The Auto Connect on Startup check box instructs Smart Wire to automatically connect to the wire service modem, start receiving data and filing stories when the program is launched. You should normally select this option and additionally set your operating system to start Smart Wire when the machine is booted. This helps minimize interruptions in service in case of power outage or any other such event.

Stories Saved To

This specifies where the stories are saved. If you are connecting to a Smart Connection Enterprise database then check that check box. If you are storing stories in a folder system then check that check box. Note that you may check both of these and use the folder stories as a backup to Enterprise.

The Raw Capture check box should never be used except to allow a technician to capture raw, unparsed, data into a file for testing purposes. This type of data can be very useful in developing parsing scripts for different wire services. See Appendix A for more details.

Wire Text Encoding

Tells the program what encoding system is being used by the wire service. This popup should be set based wire service you will be connecting to. This is very important and if not properly set can result in garbage characters in your stories.

Story Delimiters

Allows the program to understand what sequence of codes defines the beginning and ending of a story. These fields should be set according to the specific wire service you will be connecting to although the examples given are quite common. The numbers to be entered are the decimal codes of the required characters separated by commas.

Choose Service Script

The Script file is used to parse the incoming story into its text and associated meta data (category, length, slug line, etc.). Since different wire services store this data

differently, the Script allows the ability to easily accommodate these variations. Scripts are available for many common wire services but can also be developed for less common ones. See Appendix A for more information on creating/modifying Script files.

Each copy of Smart Wire includes one Script for the service indicated in the sales order.

The “Choose Service Script” button allows you to set the path to the appropriate Script file for the service in use.

Test Service Script

Tests the Script file for syntax errors. Not normally needed with provided scripts. Again, see Appendix A for more information.

Database

The preference screen is where the information to connect to the Smart Connection Enterprise database, Publication and Issue are chosen. It is also the screen where the link between the wire service categories and the database Sections is made.

Note that documents are always saved as InCopy documents when saving to an Enterprise server.

Preferences

Misc. Folders **DataBase**

DataBase Server: iMac42

User Name: woodwing

Password: **

Publication: WW News

Issue: 2nd Issue

Status: Draft text

Log In

Editions

C	Category	Section
*	Default	News
s	Sports	Sport
f	financial	News

Add

Delete

Save Cancel

Database Server

Select the desired server from the popup list.

User Name

Enter the user name. You will probably want to create a special one to be used only by Smart Wire. It must have appropriate privileges to save the stories.

Password

Enter the user's password.

Log In

When you have selected the server and the log in information, click on this button to actually log in to the server. This will then load all of the Publications, Issue, Section and Edition (4.2 and above servers) information into the window controls. You will also need to do this each time you enter this screen if you desire to change any of the settings. Once Logged In, you can optionally use this button to Log Out although that will be done for you automatically when you Save or Cancel the Preferences window.

Publication

Use this popup to chose the Publication.

Issue

Use this popup to chose the desired Issue.

Editions (4.2 or new servers only)

Select which Edition(s) you want these stories to be assigned to.

Making the Connection between Category Codes and Sections

At this point you are ready to make the connections between wire service category codes and sections within the Publication and Issue you have decided to use for this service.

The list of entries contains three columns. The first column (abbreviated C for category) contains the category character, the second column contains your description (or the wire service's description) of what this category is, and the third column is the Section that you link the category to.

When you enter this screen for the first time, the only thing in you will see in the list is the Default entry. The category code column contains a bullet character to denote that it is the default entry. Any stories received with a category character not defined in this list will go to the Section designated in this default entry. Always choose a section for this.

Next, click on the Add button. This will add an entry into the list with all columns blank except the Section set to None. When you click on this entry, or any other entry, you will be able to enter data for the three columns by using the text fields above the first two columns and the pop up for the Section column.

Add as many entries as make sense for your work flow. Note that multiple entries in the list can go to the same Section.

You can change entries in the list just by clicking on them and modifying the fields.

You can delete entries by selecting them in the list and clicking on the Delete button.

Folders

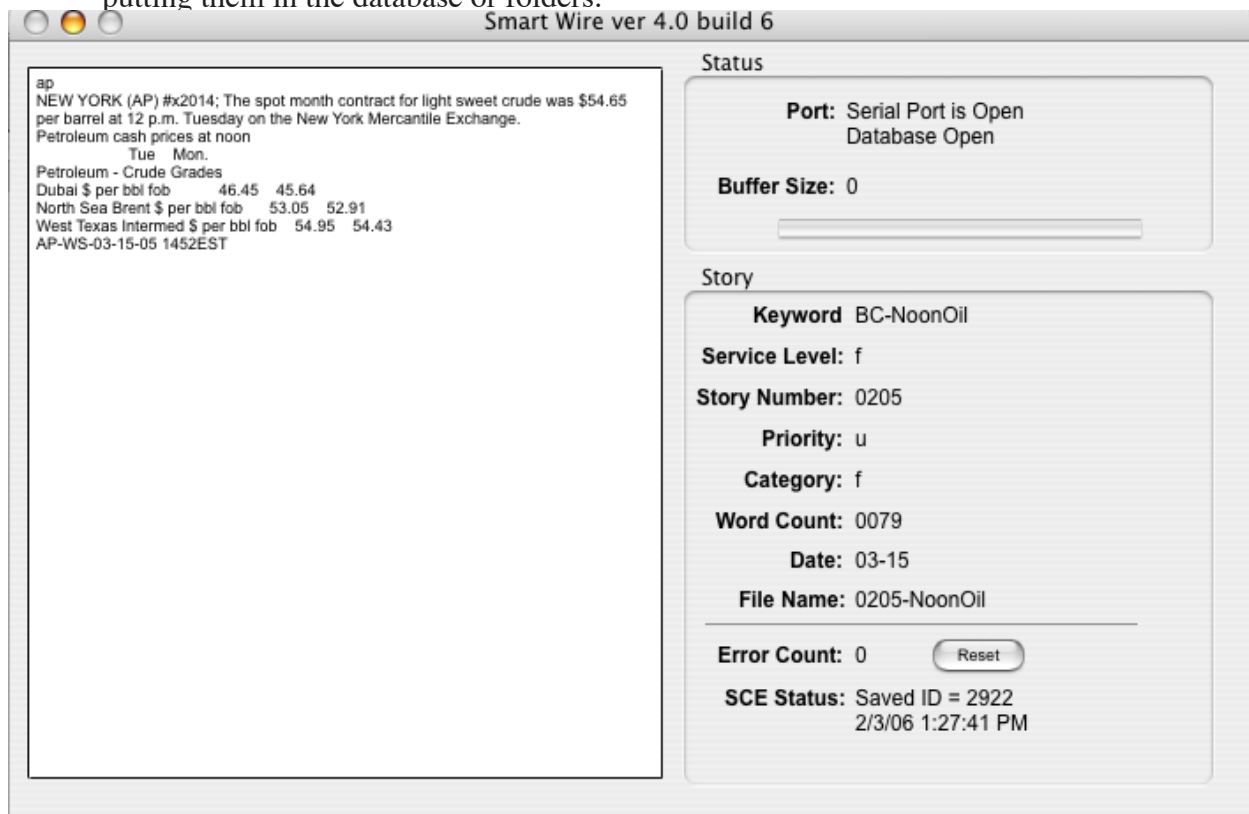
Setting Folders is very similar to the Database option but instead of “mapping” wire categories to Enterprise Sections, you are mapping those categories to folders on a file server or other disk. Due to this similarity no further description is required.

Running Smart Wire

Startup

Once all connections are made and the preferences are set, you are ready to start capturing data.

Go to the File menu and select Connect. This tells Smart Wire to connect to the Enterprise database and the chosen serial port then to start capturing wire stories and putting them in the database or folders.



Once the program starts receiving and saving stories from the wire service you will see the text and parameters/meta data similar to what is displayed above. Please note that there can be long pauses between stories and it may take some time before the first story is received.

If the Smart Wire detects an error, the “Error Count” will increment and the program will create a folder called Wire Errors in the same folder as the application. That folder will contain several text files which will allow a technician to diagnose the problem. Clicking the Reset button will restart the count at zero but does not remove the files from the errors folder. Please check this counter on a regular basis and if it is not zero contact your support line and be prepared to send in the contents of the Wire Errors folder.

If you need to change some of your Preferences, you must use the File Disconnect menu to stop the service. Then you can access the Preferences menu and modify as required, Save and reconnect using the File menu.

Appendix A Scripts

One of the most powerful features of Smart Wire is its implementation of customizable Scripts. Since the wire service formats vary, as do individual customers needs, implementation of Scripts provides the flexibility to meet those needs.

Following is the basic information needed to write and maintain these Scripts. The sample Scripts provided with the application can also serve as examples.

Scripts are merely a text file that includes computer code to parse the raw story as it is received from the wire service. Once Smart Wire detects both the Start and End Story Delimiters as defined on the Misc Preference window, it passes all the “contained” data to the Script. The Script in turn then parses the data and returns the actual story and its associated meta data back to the main program where it is then saved appropriately.

It is very important for the Script to be specifically designed for the wire service being used. If is not or it contains logic errors then unpredictable results will occur. Do not change or implement your own Script unless you fully understand the wire service format specification.

Scripts are written in RealBasic (www.realsoftware.com). It was chosen because of its robust but easy to understand syntax and feature set. Additional documentation is available on the RealSoftware web site in the Language Reference manual. Refer to the section on “RBScript”.

A complete primer in writing Scripts is beyond this document but there are a few specifics to point out.

- Scripts are just text files that can be edited with any basic text editor. They are compiled automatically by Smart Wire as needed but are also compiled when you click the Test Script button on the Preferences Misc window.
- When the Script is compiled, it will display any syntax errors along with the line number. Smart Wire cannot operate if the Script includes any such errors.
- Just because a Script compiles does not mean it will properly parse the wire data. It is very important to check the logic over carefully and make sure it complies with the format specification used by the wire service.
- In addition to the information included here, the Script file that was included with your copy of Smart Wire, includes basic documentation for implementation.
- The “input” command found near the start of the Script is how the data is passed from the main program. The “print” commands found near the end of the script are how the parsed data is passed back to the main program. These lines of code should not typically be changed.

- Story “header” and “footer” is the XML text required to make the text into an InCopy document. If they are not explicitly created in the Script then the main program will apply minimal code to create the document.
- Multipart InCopy documents for use with Smart Layout can be created. This allows for headlines, captions and other specific text to be properly styled when placed on an InDesign page.

The best way to test a Script is to capture some “Raw” wire data (see Preferences Misc). The resulting file will server two purposes:

1. A hex editor can be used to view the exact format of the incoming data and make sure it complies with the specification or to develop one if none is available.
2. You can then repeatedly feed that raw file back into Smart Wire just like it was being received via the service modem and serial port, thus testing your Script. This is done by holding the shift key down before mousing-over and invoking the File Connect menu. When you do that Smart Wire will not open the serial port as it normally does, but instead will display a “open file” dialog where you can select the raw file and have it “feed” into the program.

If you use this method it is suggest that you capture a significant amount of Raw data, probably 100 stories or more. This is to insure that you have a good sampling of the data with hopefully examples of all the variations that might normally occur.

The following strings can be passed from the Script to Smart Wire:

- as actual story
- kw keyword
- fn file name
- wc word count
- pr priority
- ca category
- sl service level
- sn story number
- sd story date
- sh story header (if empty SW puts minimal xml to make IC doc)
- sf story footer (if empty SW puts minimal xml to make IC doc)
- fi format ID (not implemented)
- vf version field (not implemented)
- rf reference field (not implemented)

These Special functions can be used in the Script to aid in debugging and parsing the story:

- **EncodeMe(decimal)** Returns character of ‘decimal’ encoded in the selected encoding
- **ScriptDate(format)** Returns current system date and or time
 - Available formats (returns strings)
 - AbbreviatedDate Wed, Dec 31, 1999

Day
 DayOfWeek 1= sunday, 7 = Saturday
 DayOfYear 1-365
 Hour
 LongDate Wednesday, December 31, 1999
 LongTime 2:32:20 PM
 Minute
 Month
 Second
 ShortDate 12/31/01
 ShortTime 2:32 PM
 SqlDate YYYY-MM-DD
 SqlDateTime YYYY-MM-DD HH:MM:SS
 TotalSeconds since Jan 1, 1904
 WeekOfYear
 Year

- **ScriptBeep** for debugging just ‘beeps’
- **ScriptMsgBox(string)** for debugging displays message in OK dialog
- **ScriptSavetoFile(string, file name, separator)** debugging saves “string” to file
- **WhatAmI(string)** returns encoding name of string passed
- **IncludeFile(pathstring)** returns the text of the file located in “pathstring”. Useful for including large amounts of boilerplate text in the story.

RB Script Syntax

Operators:

+, -, *, /, \, Mod, <, =, >, <=, >=, <>.

And, Not, Or.

Comments are supported: ‘, //, and REM.

Data Types

Integer
 Single
 Double
 Boolean
 String
 Color
 Object
 Variant

Control Structures

Function...

Sub...

For... Next

Do... Loop

If... Then... End If

Select Case... End Select

While...Wend

Return statement

Functions

Comment

Abs(Double) As Double

Acos(Double) As Double

Asc(String) As Integer

AscB(String) As Integer

Asin(Double) As Double

Atan(Double) As Double

Atan2(Double, Double) As Double

ByRef keyword

ByVal keyword

CDBl(String) As Double

Identical to Val().

Ceil(Double) As Double

Chr(Double) As String

ChrB(Double) As String

CMY

Color

Const name = value

Cos(Double) As Double

CountFields(String, String) As Integer

CStr(Double) as String

Dim

Allows you to declare multiple variables with different types rather than only multiple variables of the same type. Empty arrays can be created by specifying -1 elements.

Do...Loop

Exit

Exp(Double) As Double

False

Floor(Double) As Double

For...Next

Allows floating point loop counters and step values.

Format(Double, String) As String

Function

Functions can be called before they are defined.

Goto keyword

Hex(Integer) As String
HSV
If...Then...Else
InStr(Integer, String, String) As Integer
InStrB(Integer, String, String) As Integer
IsA operator
Can be used with Object; it will return True for any non-null object.
Left(String, Integer) As String
LeftB(String, Integer) As String
Len(String) As Integer
LenB(String) As Integer
Log(Double) As Double
Lowercase(String) As String
LTrim(String) As String
Max(Double, Double) As Double
Me
Microseconds As Double
Mid(String, Integer, Integer) As String
MidB(String, Integer, Integer) As String
Min(Double, Double) As Double
Nil
NthField(String, String, Integer) As String
Oct(Integer) As String
Pow(Double, Double) As Double
Redim
Rem
Replace(String, String, String) As String
ReplaceB(String, String, String) As String
ReplaceAll(String, String, String) As String
ReplaceAllB(String, String, String) As String
RGB
Right(String, Integer) As String
RightB(String, Integer) As String
Rnd As Double
Round(Double) As Double
RTrim(String) As String
Select Case
Self
Sin(Double) As Double
Sqrt(Double) As Double
Str(Double) As String
StrComp(String, String, Integer) As Integer
Sub
Tan(Double) As Double
Ticks as Integer
Titlecase(String) As String

Trim(String) As String
True
Ubound(array) As Integer
Supports second parameter for multi-dimensional arrays.
Uppercase(String) As String
Val(String) As Double
Scientific format not recognized.
While...Wend

RB Script Errors

Compiler error numbers returned in errorNumber are shown below:

- 1 Syntax does not make sense.
- 2 Type mismatch.
- 3 Select Case does not support that type of expression.
- 4 The compiler is not implemented (obsolete).
- 5 The parser's internal stack has overflowed.
- 6 Too many parameters for this function.
- 7 Not enough parameters for this function call.
- 8 Wrong number of parameters for this function call.
- 9 Parameters are incompatible with this function.
- 10 Assignment of an incompatible data type.
- 11 Undefined identifier.
- 12 Undefined operator.
- 13 Logic operations require Boolean operands.
- 14 Array bounds must be integers.
- 15 Can't call a non-function.
- 16 Can't get an element from something that isn't an array.
- 17 Not enough subscripts for this array's dimensions.
- 18 Too many subscripts for this array's dimensions.
- 19 Can't assign an entire array.
- 20 Can't use an entire array in an expression.
- 21 Can't pass an expression as a ByRef parameter.
- 22 Duplicate identifier.
- 23 The backend code generator failed.
- 24 Ambiguous call to overloaded method.
- 25 Multiple inheritance is not allowed.
- 26 Cannot create an instance of an interface.
- 27 Cannot implement a class as though it were an interface.
- 28 Cannot inherit from something that is not a class.
- 29 This class does not fully implement the specified interface.
- 30 Event handlers cannot live outside of a class.
- 31 It is not legal to ignore the result of a function call.
- 32 Can't use "Self" keyword outside of a class.
- 33 Can't use "Me" keyword outside of a class.
- 34 Can't return a value from a Sub.

35 An exception object required here.

36-39 Obsolete.

40 Destructors can't have parameters.

41 Can't use "Super" keyword outside of a class.

42 Can't use "Super" keyword in a class that has no parent.

Appendix B Release Notes

Version 4.2.0 Build 8

New “IncludeFile()” added to the available Script functions.

Version 4.2.0 Build 7

Fix Bug which would not properly save System Default as a Text Encoding Preference

Version 4.2.0 Build 6

New “Wire Text Encoding” Preference now includes all code pages supported by the underlying operating system.

Version 4.2.1 Build 5

New Supports HTTPS simply by including the ‘S’ in URL

New Supports Authorization & UnAuthorization using a proxy server.

Version 4.2.0 Build 4

New Mac Universal Binary

New SCEnt 4.2.x support